

# Esercizi Cache



## organizzazione e tecniche di allocazione

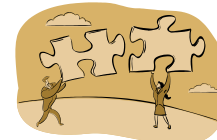
**Es4c:** Con riferimento alle sequenze mostrate nell'es4a, supponendo di avere una cache ad associazione diretta in grado di memorizzare 8 parole, quale fra le seguenti dimensioni di blocco

- a) 1 parola
- b) 2 parole
- c) 4 parole

è la più conveniente (minimizza il numero di miss) ?

**Es4d:** Ripetere l'esercizio 4a nel caso di una cache ad associazione a 2 vie in grado di memorizzare 16 blocchi, ognuno costituito da 1 parola. Usare la politica FIFO per il rimpiazzo dei blocchi.

### Soluz. 4c:



- trattandosi di una cache con associazione diretta, l'indirizzo di memoria centrale deve essere suddiviso nei campi etichetta, linea, e parola, ed in particolare:

- a) blocco costituito da una sola parola:  
il campo parola ha 0 bit; il campo linea sarà costituito da 3 bit in quanto occorre indirizzare 8 ( $= 2^3$ ) linee (blocchi) di cache; il campo tag sarà quindi costituito da  $(32 - 3 - 0) = 29$  bit;
- b) blocco costituito da 2 parole:  
il campo parola ha 1 bit ( $2^1$  parole); il campo linea sarà costituito da 2 bit in quanto occorre indirizzare 4 ( $= 2^2$ ) linee (blocchi) di cache; il campo tag sarà quindi costituito da  $(32 - 2 - 1) = 29$  bit;
- c) blocco costituito da 4 parole:  
il campo parola ha 2 bit ( $2^2$  parole); il campo linea sarà costituito da 1 bit in quanto occorre indirizzare 2 ( $= 2^1$ ) linee (blocchi) di cache; il campo tag sarà quindi costituito da  $(32 - 1 - 2) = 29$  bit.

Blocco di 1 parola

	Sequenza 1	h/m	cache
	[ tag   linea]		
1	0000000000000000000000000000000000001	miss	blocco 1 <sub>dec</sub> in linea 001
2	0000000000000000000000000000010000110	miss	blocco 134 <sub>dec</sub> in linea 110
3	0000000000000000000000000000011010100	miss	blocco 212 <sub>dec</sub> in linea 100
4	0000000000000000000000000000000000001	hit	
5	0000000000000000000000000000010000111	miss	blocco 135 <sub>dec</sub> in linea 111
6	0000000000000000000000000000011010101	miss	blocco 213 <sub>dec</sub> in linea 101
7	0000000000000000000000000000010100010	miss	blocco 162 <sub>dec</sub> in linea 010
8	0000000000000000000000000000010100001	miss	blocco 161 <sub>dec</sub> in linea 001 [ 1 <sub>dec</sub> out]
9	0000000000000000000000000000000000010	miss	blocco 2 <sub>dec</sub> in linea 010 [162 <sub>dec</sub> out]
10	00000000000000000000000000000101100	miss	blocco 44 <sub>dec</sub> in linea 100 [212 <sub>dec</sub> out]
11	00000000000000000000000000000101001	miss	blocco 41 <sub>dec</sub> in linea 001 [161 <sub>dec</sub> out]
12	0000000000000000000000000000011011101	miss	blocco 221 <sub>dec</sub> in linea 101 [213 <sub>dec</sub> out]

Blocco di 1 parola

	Sequenza 2	h/m	cache
	[ tag   linea]		
1	00000000000000000000000000000000000100	miss	blocco 4 <sub>dec</sub> in linea 100
2	00000000000000000000000000000110101110	miss	blocco 214 <sub>dec</sub> in linea 110
3	0000000000000000000000000000010101111	miss	blocco 175 <sub>dec</sub> in linea 111
4	00000000000000000000000000000110101110	hit	
5	00000000000000000000000000000000000100	hit	
6	000000000000000000000000000001010100	miss	blocco 84 <sub>dec</sub> in linea 100 [ 4 <sub>dec</sub> out]
7	0000000000000000000000000000010000001	miss	blocco 65 <sub>dec</sub> in linea 001
8	00000000000000000000000000000101011110	miss	blocco 174 <sub>dec</sub> in linea 110 [214 <sub>dec</sub> out]
9	0000000000000000000000000000010000000	miss	blocco 64 <sub>dec</sub> in linea 000
10	000000000000000000000000000001101001	miss	blocco 105 <sub>dec</sub> in linea 001 [ 65 <sub>dec</sub> out]
11	000000000000000000000000000001010101	miss	blocco 85 <sub>dec</sub> in linea 101
12	0000000000000000000000000000011010111	miss	blocco 215 <sub>dec</sub> in linea 111 [175 <sub>dec</sub> out]

Blocco di 2 parole    l = linea  
                               p = parola

	Sequenza 1	h/m	cache
	[ tag   1   p ]		
1	000000000000000000000000000000000001	miss	blocco $1_{dec}/2 = 0_{dec}$ in linea 00
2	0000000000000000000000000000010000110	miss	blocco $134_{dec}/2 = 67_{dec}$ in linea 11
3	0000000000000000000000000000011010100	miss	blocco $212_{dec}/2 = 106_{dec}$ in linea 10
4	000000000000000000000000000000000001	hit	
5	0000000000000000000000000000010000111	hit	
6	0000000000000000000000000000011010101	hit	
7	0000000000000000000000000000010100010	miss	blocco $162_{dec}/2 = 81_{dec}$ in linea 01
8	0000000000000000000000000000010100001	miss	blocco $161_{dec}/2 = 80_{dec}$ in linea 00 a
9	0000000000000000000000000000000000010	miss	blocco $2_{dec}/2 = 1_{dec}$ in linea 01 b
10	000000000000000000000000000000101100	miss	blocco $44_{dec}/2 = 22_{dec}$ in linea 10 c
11	000000000000000000000000000000101001	miss	blocco $41_{dec}/2 = 20_{dec}$ in linea 00 d
12	000000000000000000000000000001101101	miss	blocco $221_{dec}/2 = 110_{dec}$ in linea 10 e

- a = [  $0_{dec}$  out ]
- b = [  $81_{dec}$  out ]
- c = [  $106_{dec}$  out ]
- d = [  $80_{dec}$  out ]
- e = [  $22_{dec}$  out ]

Blocco di 2 parole    l = linea  
                               p = parola

	Sequenza 2	h/m	cache
	[ tag   1   p ]		
1	00000000000000000000000000000000000100	miss	blocco $4_{dec}/2 = 2_{dec}$ in linea 10
2	0000000000000000000000000000011010110	miss	blocco $214_{dec}/2 = 107_{dec}$ in linea 11
3	0000000000000000000000000000010101111	miss	blocco $175_{dec}/2 = 87_{dec}$ in linea 11 a
4	0000000000000000000000000000011010110	miss	blocco $214_{dec}/2 = 107_{dec}$ in linea 11 b
5	00000000000000000000000000000000000100	hit	
6	0000000000000000000000000000001010100	miss	blocco $84_{dec}/2 = 42_{dec}$ in linea 10 c
7	0000000000000000000000000000001000001	miss	blocco $65_{dec}/2 = 32_{dec}$ in linea 00
8	0000000000000000000000000000010101110	miss	blocco $174_{dec}/2 = 87_{dec}$ in linea 11 d
9	0000000000000000000000000000001000000	hit	
10	0000000000000000000000000000001101001	miss	blocco $105_{dec}/2 = 52_{dec}$ in linea 00 e
11	0000000000000000000000000000001010101	hit	
12	0000000000000000000000000000011010111	miss	blocco $215_{dec}/2 = 107_{dec}$ in linea 11 f

- a = [  $102_{dec}$  out ]
- b = [  $87_{dec}$  out ]
- c = [  $2_{dec}$  out ]
- d = [  $102_{dec}$  out ]
- e = [  $32_{dec}$  out ]
- f = [  $87_{dec}$  out ]



Blocco di 4 parole  $l = \text{linea}$   
 $p = \text{parola}$

	Sequenza 1	h/m	cache
	[ tag   l   p ]		
1	0000000000000000000000000000001	miss	blocco $1_{dec}/4 = 0_{dec}$ in linea 0
2	0000000000000000000000000000110	miss	blocco $134_{dec}/4 = 33_{dec}$ in linea 1
3	0000000000000000000000000000110100	miss	blocco $212_{dec}/4 = 53_{dec}$ in linea 1 a
4	0000000000000000000000000000001	hit	
5	00000000000000000000000000001000111	miss	blocco $135_{dec}/4 = 33_{dec}$ in linea 1 b
6	00000000000000000000000000001101010	miss	blocco $213_{dec}/4 = 53_{dec}$ in linea 1 c
7	000000000000000000000000000010100010	miss	blocco $162_{dec}/4 = 40_{dec}$ in linea 0 d
8	00000000000000000000000000001010001	hit	
9	00000000000000000000000000000010	miss	blocco $2_{dec}/4 = 0_{dec}$ in linea 0 e
10	0000000000000000000000000000101100	miss	blocco $44_{dec}/4 = 11_{dec}$ in linea 1 f
11	0000000000000000000000000000101001	miss	blocco $41_{dec}/4 = 10_{dec}$ in linea 0 g
12	00000000000000000000000000001101101	miss	blocco $221_{dec}/4 = 55_{dec}$ in linea 1 h

a = [  $33_{dec}$  out ]     f = [  $53_{dec}$  out ]  
b = [  $53_{dec}$  out ]     g = [  $0_{dec}$  out ]  
c = [  $33_{dec}$  out ]     h = [  $11_{dec}$  out ]  
d = [  $0_{dec}$  out ]  
e = [  $40_{dec}$  out ]



Blocco di 4 parole  $l = \text{linea}$   
 $p = \text{parola}$

	Sequenza 2	h/m	cache
	[ tag   l   p ]		
1	0000000000000000000000000000100	miss	blocco $4_{dec}/4 = 1_{dec}$ in linea 1
2	000000000000000000000000000011010110	miss	blocco $214_{dec}/4 = 53_{dec}$ in linea 1 a
3	000000000000000000000000000010101111	miss	blocco $175_{dec}/4 = 43_{dec}$ in linea 1 b
4	000000000000000000000000000011010110	miss	blocco $214_{dec}/4 = 53_{dec}$ in linea 1 c
5	000000000000000000000000000000100	miss	blocco $4_{dec}/4 = 1_{dec}$ in linea 1 d
6	00000000000000000000000000001010100	miss	blocco $84_{dec}/4 = 21_{dec}$ in linea 1 e
7	00000000000000000000000000001000001	miss	blocco $65_{dec}/4 = 16_{dec}$ in linea 0
8	000000000000000000000000000010101110	miss	blocco $174_{dec}/4 = 43_{dec}$ in linea 1 f
9	00000000000000000000000000001000000	hit	
10	00000000000000000000000000001101001	miss	blocco $105_{dec}/4 = 26_{dec}$ in linea 0 g
11	00000000000000000000000000001010101	miss	blocco $85_{dec}/4 = 21_{dec}$ in linea 1 h
12	000000000000000000000000000011010111	miss	blocco $215_{dec}/4 = 53_{dec}$ in linea 1 i

a = [  $1_{dec}$  out ]     f = [  $21_{dec}$  out ]  
b = [  $53_{dec}$  out ]     g = [  $16_{dec}$  out ]  
c = [  $43_{dec}$  out ]     h = [  $43_{dec}$  out ]  
d = [  $53_{dec}$  out ]     i = [  $21_{dec}$  out ]  
e = [  $1_{dec}$  out ]

Riassumendo i risultati ottenuti abbiamo:

Dimensione blocco	Numero totale miss	Numero totale hit	Migliore
1	21	3	X
2	18	6	
4	21	3	

## Esercizi Cache



### organizzazione e tecniche di allocazione

**Es4c:** Con riferimento alle sequenze mostrate nell'es4a, supponendo di avere una cache ad associazione diretta in grado di memorizzare 8 parole, quale fra le seguenti dimensioni di blocco

- a) 1 parola
- b) 2 parole
- c) 4 parole

è la più conveniente (minimizza il numero di miss) ?

**Es4d:** Ripetere l'esercizio 4a nel caso di una cache ad associazione a 2 vie in grado di memorizzare 16 blocchi, ognuno costituito da 1 parola. Usare la politica FIFO per il rimpiazzo dei blocchi.





	Sequenza 2	h/m	cache
	[ tag   set ]		
1	0000000000000000000000000000100	miss	blocco 4 <sub>dec</sub> in set 100[0]
2	000000000000000000000000000011010110	miss	blocco 214 <sub>dec</sub> in set 110[0]
3	000000000000000000000000000010101111	miss	blocco 175 <sub>dec</sub> in set 111[0]
4	000000000000000000000000000011010110	hit	
5	000000000000000000000000000000000000	hit	
6	000000000000000000000000000001010100	miss	blocco 84 <sub>dec</sub> in set 100[1]
7	000000000000000000000000000001000001	miss	blocco 65 <sub>dec</sub> in set 001[0]
8	000000000000000000000000000001010110	miss	blocco 174 <sub>dec</sub> in set 110[1]
9	000000000000000000000000000001000000	miss	blocco 64 <sub>dec</sub> in set 000[0]
10	000000000000000000000000000001101001	miss	blocco 105 <sub>dec</sub> in set 001[1]
11	000000000000000000000000000001010101	miss	blocco 85 <sub>dec</sub> in set 101[0]
12	000000000000000000000000000001101011	miss	blocco 215 <sub>dec</sub> in set 111[1]

set 001[0] si riferisce alla linea 0 del set 001  
 set 001[1] si riferisce alla linea 1 del set 001



**Es5:** Sia data la seguente sequenza di indirizzi in lettura (l) o scrittura (s) emessi dalla CPU:

	Indirizzo	l/s	dato scritto (in esadecimale)
1	000100000000	l	
2	000100001000	l	
3	000100001100	s	B1
4	000100001100	l	
5	000100010000	s	B4
6	000100010000	l	
7	000100010100	s	B7

Si assuma che la dimensione di parola coincida con un byte, e la presenza di una cache di ampiezza 16B, dimensione di blocco 4B, inizialmente vuota, e ad associazione a 2 vie (con politica di rimpiazzo LRU e politica di scrittura write-through).

Si assuma che la memoria abbia il contenuto esadecimale mostrato di seguito:

ind	byte	ind	byte	ind	byte	ind	byte
100	0C	101	00	102	07	103	02
104	00	105	00	106	00	107	00
108	AE	109	13	10A	A1	10B	23
10C	A1	10D	42	10E	90	10F	75
110	B9	111	16	112	00	113	00
114	0A	115	07	116	03	117	71

ind = indirizzo

**Si mostri come sia il contenuto della cache che il contenuto della memoria cambia.**

### Soluzione:

Poiché un blocco è costituito da 4B, e la cache è di 16B, si avranno in cache  $16/4 = 4$  linee.

Essendo l'associatività a due linee (2 vie), la cache sarà costituita da due insiemi (set 0 e set 1) ognuno di 2 linee.

Quindi i 12 bit di indirizzo saranno suddivisi nel seguente modo:

- i 2 bit meno significativi individueranno il byte all'interno del blocco;
- il terzo bit da destra individuerà l'insieme (set 0 o set 1);
- i restanti bit costituiranno il campo tag.

Mostriamo di seguito l'evoluzione del contenuto della cache e della memoria.

Per la cache, nel caso in cui tutte e due le linee di un insieme (set) siano libere, si sceglie la linea con indirizzo minore per la allocazione (scelta arbitraria: si poteva usare un criterio diverso).

In caso di miss per una operazione di scrittura, si assume la politica "write allocate", cioè si porta prima in cache il blocco che contiene la parola da scrivere e poi si effettua la scrittura.

#### Codifica della soluzione

ind. rif. memoria	cache dati		modifica memoria mem[ind.] = cont.
	set 0	set 1	
hex	[ linea 0 ]	[ linea 2 ]	
binario	t: tag	t: tag	
	r: rif.	r: rif.	
	[ linea 1 ]	[ linea 3 ]	
	t: tag	t: tag	
	r: rif.	r: rif.	

```
100      [0C|00|07|02]
000100000000 t:000100000
r:miss
[          ]
t:
r:
```

```
108      [0C|00|07|02]
000100001000 t:000100000
r:
[AE|13|A1|23]
t:000100001
r:miss
```